IT-Security (ITS) B1

DIKU, E2022

Today’s agenda

Part 1: Crypto building blocks

Part 2: More crypto building blocks

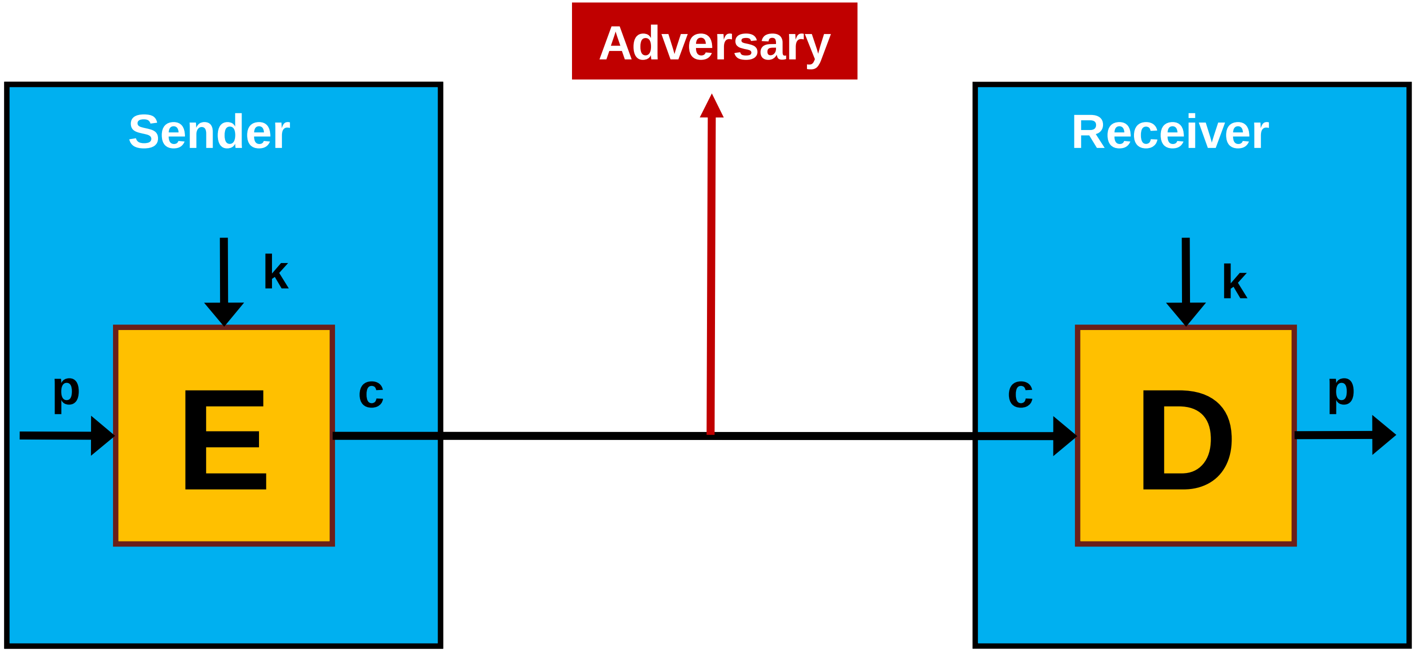
(Later: Real-world crypto protocols)

Lecture plan

Week	Date	Time	Instructor	Topic
36	05 Sep	10-12	TL	Security concepts and principles
	09 Sep	**10-12**	**TL**	**Cryptographic building blocks**
37	12 Sep	10-12	TL	Key establishment and certificate management
	16 Sep	10-12	CJ	User authentication, IAM
38	19 Sep	10-12	CJ	Operating systems security, web, browser and mail security
	23 Sep	10-12	CJ	IT security management and risk assessment
39	26 Sep	10-12	TL	Software security - exploits and privilege escalation
	30 Sep	10-12	TL	Malicious software
40	03 Oct	10-12	CJ	Firewalls and tunnels, security architecture
	07 Oct	10-12	CJ	Cloud and IoT security
41	10 Oct	10-12	TL	Intrusion detection and network attacks
	14 Oct	10-12	TL	Forensics
42				Fall Vacation - No lectures
43	24 Oct	10-12	CJ	Privacy and GDPR
	28 Oct	10-12	CJ	Privacy engineering
44	31 Oct	10-11	Guest	Special topic
		11-12	TL,CJ	Exam Q/A

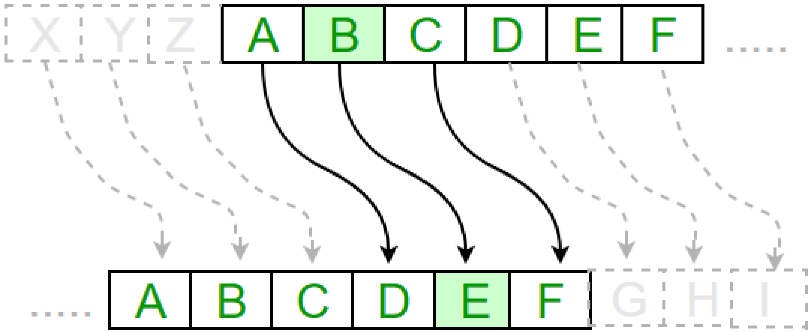
https://github.com/diku-its/its-e2022/blob/main/lectureplan2022.md

Cryptosystems



Early cryptography – Caesar cipher

Early cryptography – Caesar cipher



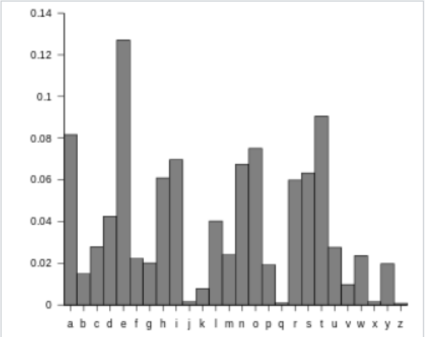
$A \rightarrow 0, B \rightarrow 1, ..., Z \rightarrow 25.$

$$E_n(x) = (x + n) \mod 26.$$

$$D_n(x) = (x - n) \mod 26.$$

Early cryptography – Caesar cipher

Algorithmic attack



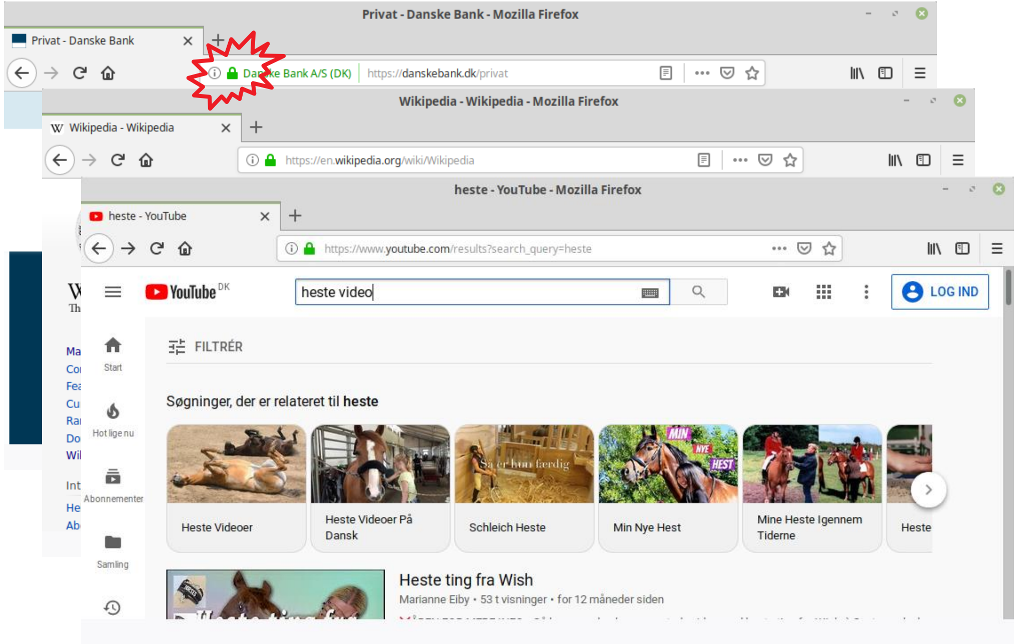
The distribution of letters in a typical sample of English language text has a distinctive and predictable shape. A Caesar shift "rotates" this distribution, and it is possible to determine the shift by examining the resultant frequency graph.

Exhaustive key search

Decryption shift	Candidate plaintext
0	exxegoexsrgi
1	dwwdfndwrqfh
2	cvvcemcvqpeg
3	buubdlbupodf
4	attackatonce
5	zsszbjzsnmbd
6	yrryaiyrmlac
...	
23	haahjrhavujl
24	gzzgiqgzutik
25	fyyfhpfytshj

Cryptography influence world events

Our goal: Secure online communication



Security goals we’re looking to achieve

Confidentiality

Integrity

Authenticity

Non-repudiation

Warm-up question

FileCrypt

“**FileCrypt** is a dynamic non-factor based quantum AI encryption hardware solution.

Developed by our cryptographic experts and hardwired into a tamper-resistant USB token.

Plug the token into your PC, start the program and encrypt the files you need to protect”

What problems do you see with this solution?

Multiple concerns

#1: “Developed by our cryptographic experts”

Should we trust proprietary crypto over public peer-reviewed time-tested crypto?

#2: “Dynamic non-factor based quantum AI”

What does that mean? Are there any academic papers that discuss this concept?

#3: “Plug the token into your PC”

Can anyone do this? What if token is lost? Violates **Kerckhoffs’ Principle**

Kerckhoffs' (2nd) Principle

The security of a cryptographic algorithm must rest solely in the secrecy of its **key**, not in the secrecy of the algorithm itself

Collaries:

Assume attacker knows the algorithm
Make it available for public analysis
Protect the key!

Auguste Kerckhoffs
(1835 – 1903)

Today’s topics

Symmetric encryption/decryption

Asymmetric encryption/decryption

Digital signatures

Message authentication codes

Cryptographic hash functions

Symmetric cryptosystems

Symmetric cryptosystems

Stream ciphers

One time pad

If *k* random, |*k*| >= |*p*|, never reused, and kept secret, then then impossible to decrypt or break without knowing the key (Shannon, 1949)

Key	0	1	0	1	1	1	0	0	1	0
Plaintext	1	1	0	0	0	1	1	0	0	0
Ciphertext	1	0	0	1	1	0	1	0	1	0

$\oplus$

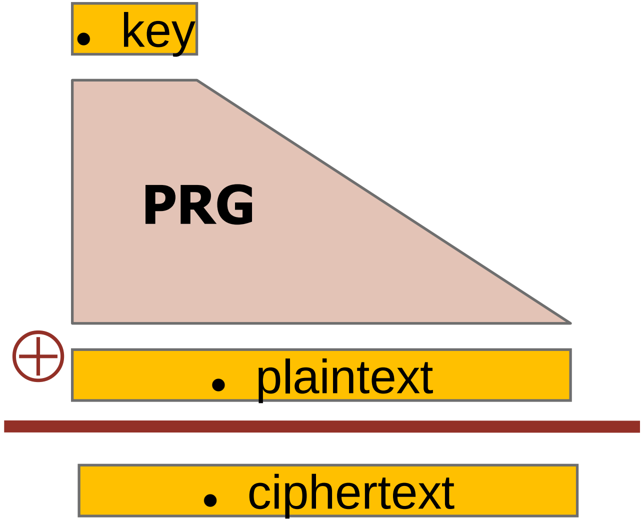
Towards modern stream ciphers

Problem

OTP key as long as plaintext

Solution

Generate pseudo random keystream



1^{st} rule of stream ciphers

Never reuse key

$$C_1 \leftarrow P_1 \oplus PRG(k)$$

$$C_2 \leftarrow P_2 \oplus PRG(k)$$

$$C_1 \oplus C_2 \rightarrow P_1 \oplus P_2$$

$$P_1 \oplus P_2 \rightarrow P_1, P_2$$

Solution: Initialisation Vector (IV)

For each message

- Generate IV
- Mix k with IV
- Generate keystream PRG(k+IV) and encrypt
- Send c and IV (in plaintext)

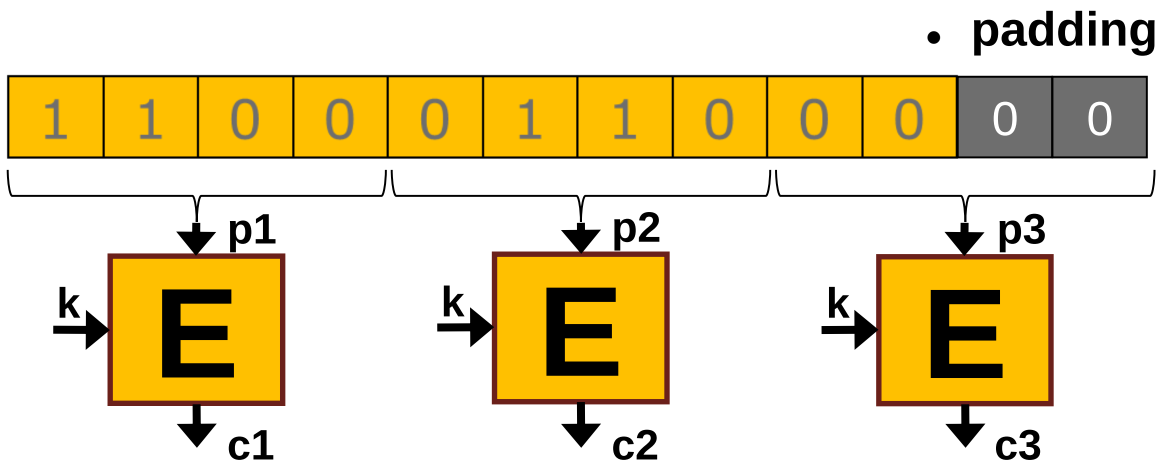
Change k before IVs run out

Stream ciphers in the wild

Block ciphers

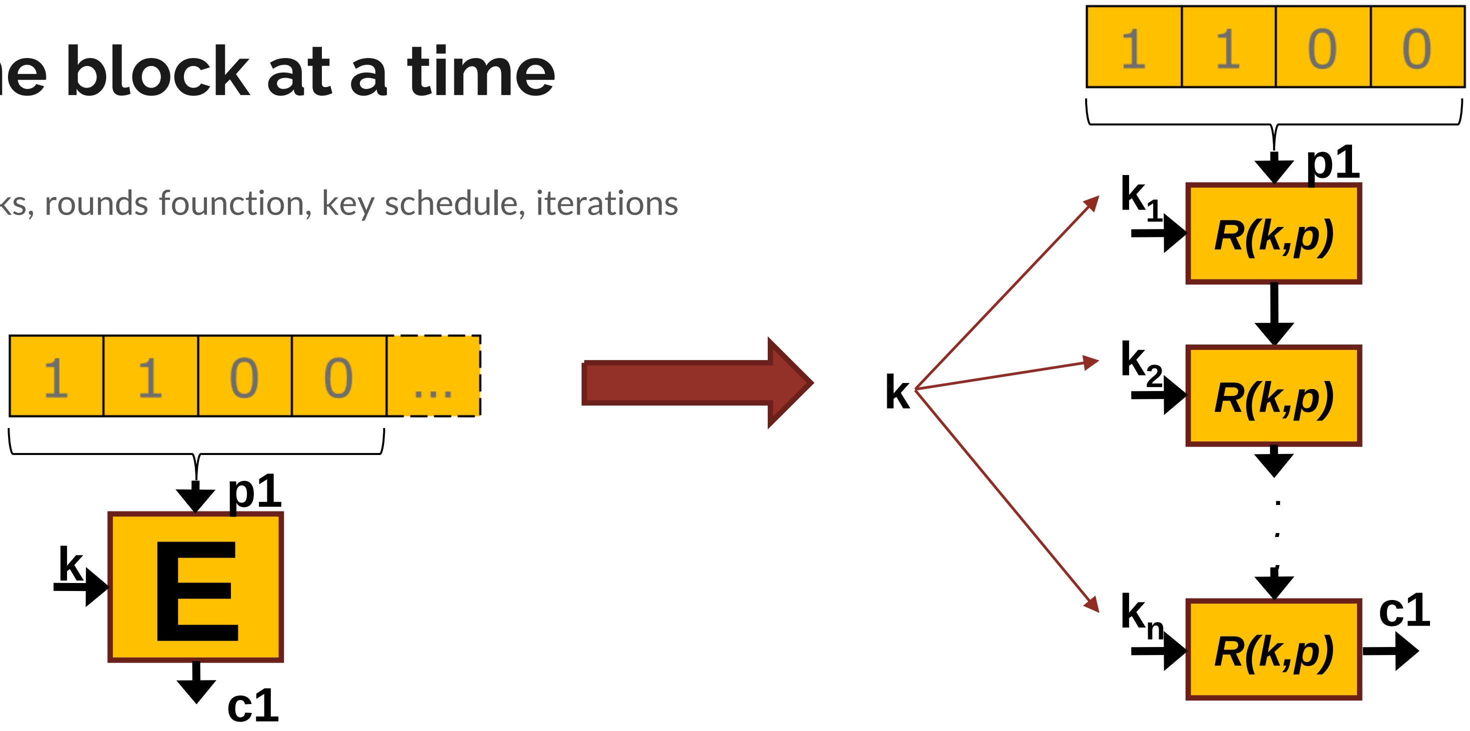
Block ciphers

One block at a time – as oppossed to one bit at a time

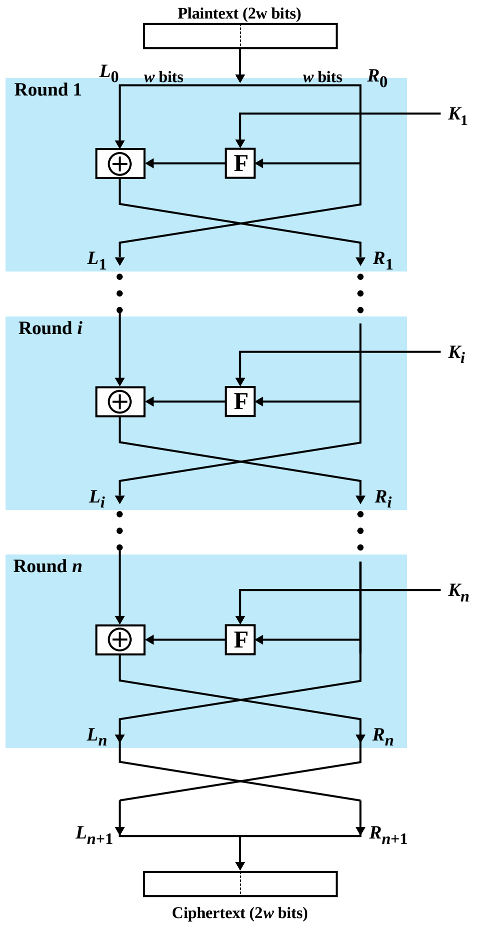


One block at a time

Blocks, rounds founction, key schedule, iterations



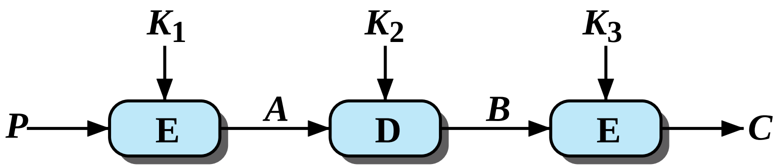
Feistel network



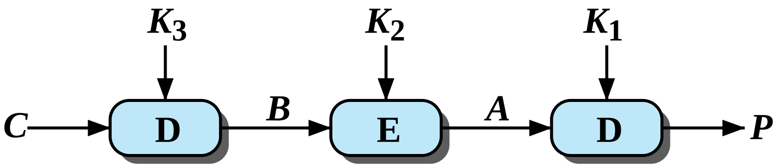
DES

DES

Key 64, block 64, rounds 16



(a) Encryption



(b) Decryption

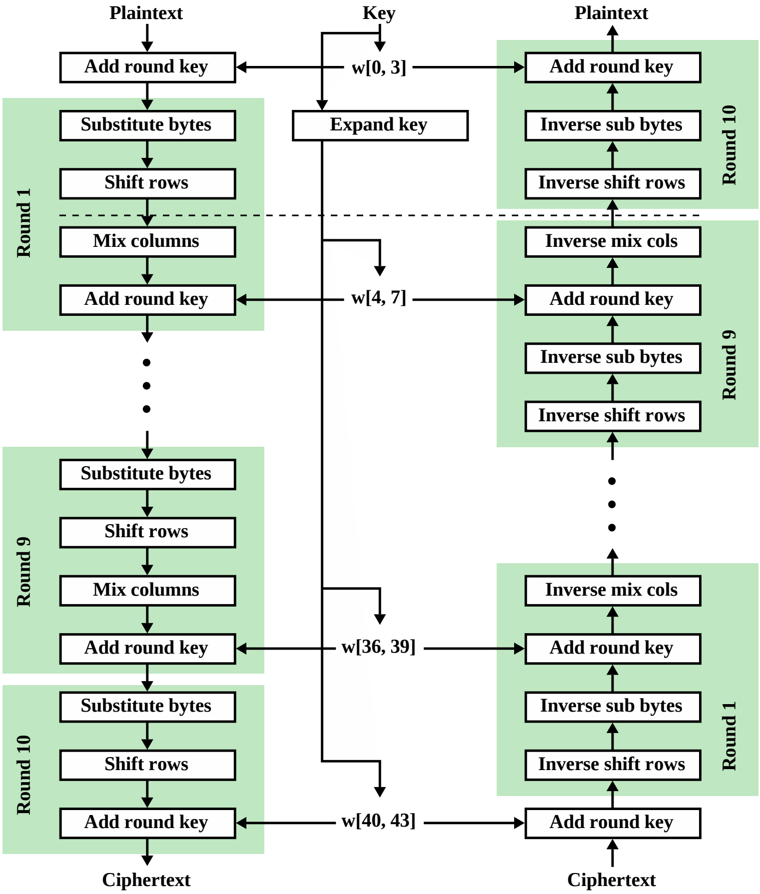
AES

AES

Keys 128/192/256

Block 128

Rounds 10/12/14

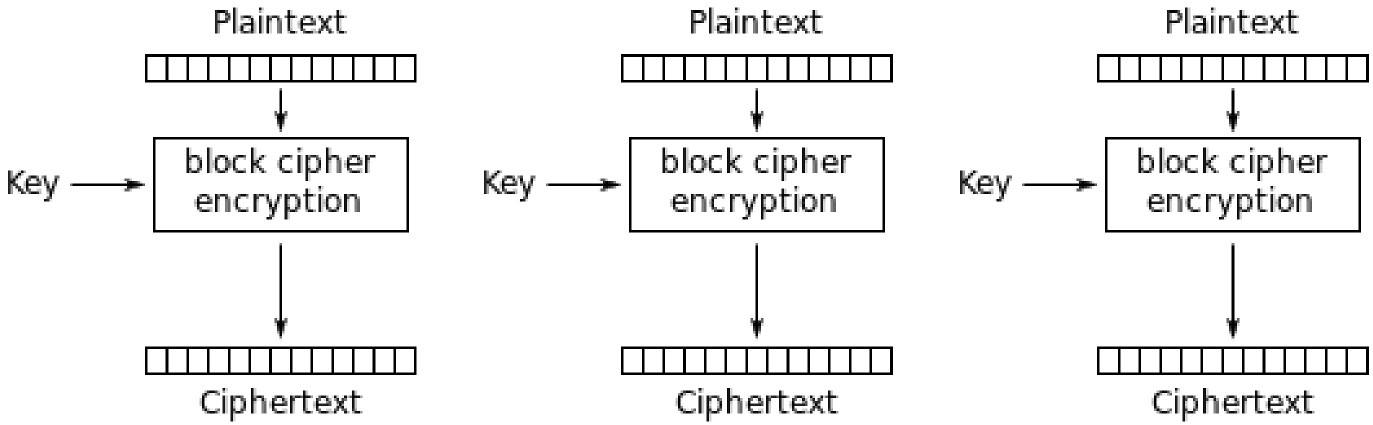


(a) Encryption

(b) Decryption

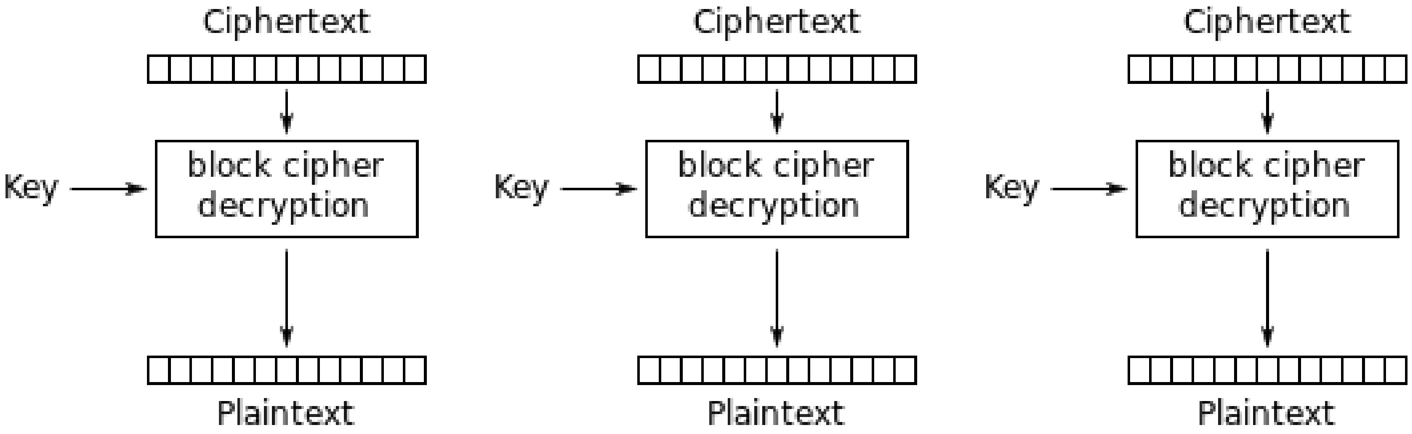
Modes of operation

Electronic Codebook (ECB)



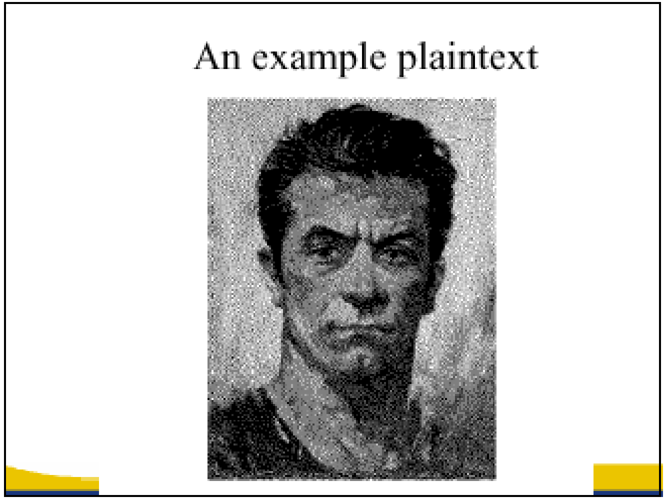
Electronic Codebook (ECB) mode encryption

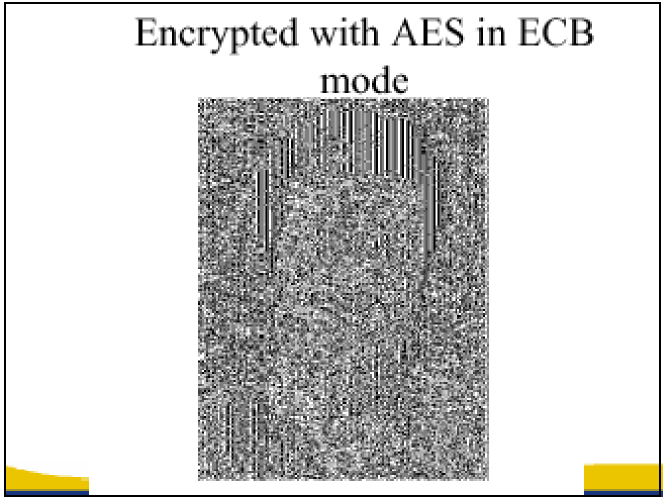
ECB decyption



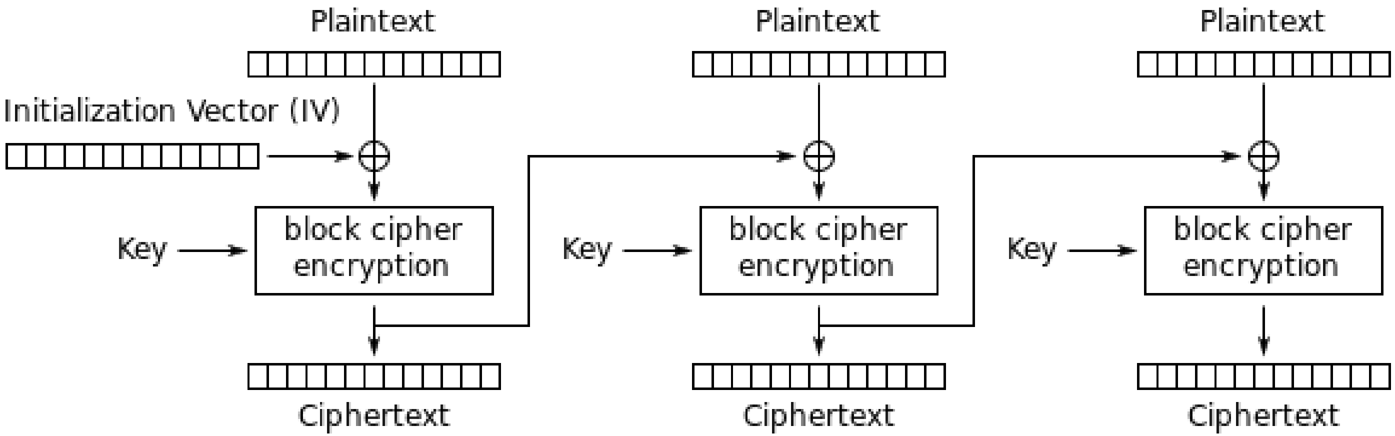
Electronic Codebook (ECB) mode decryption

If p1 = p2, then c1 = c2



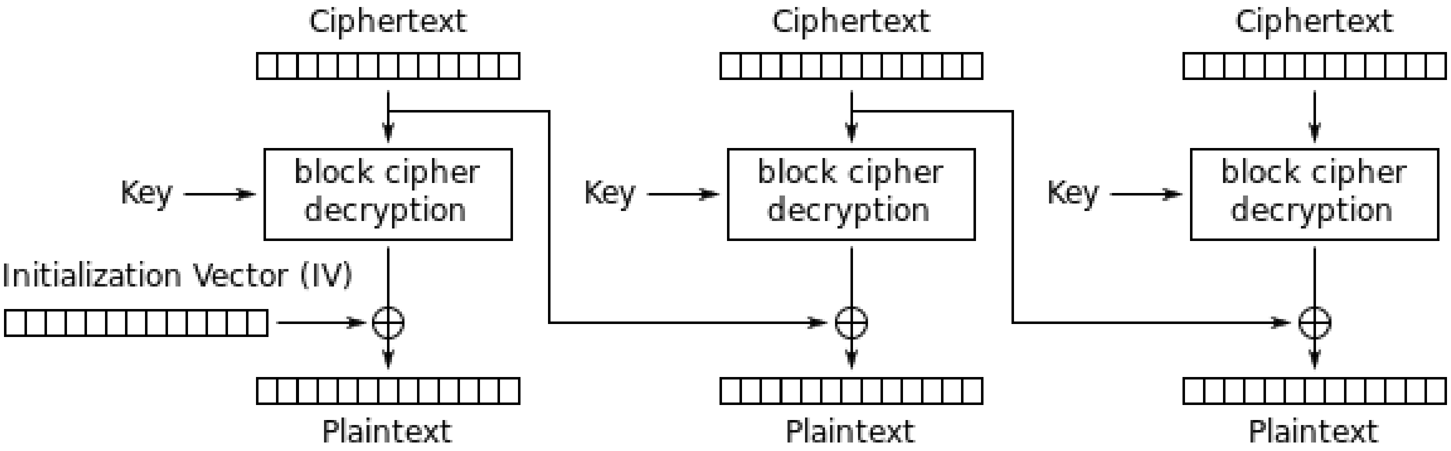


Cipher Block Chaining



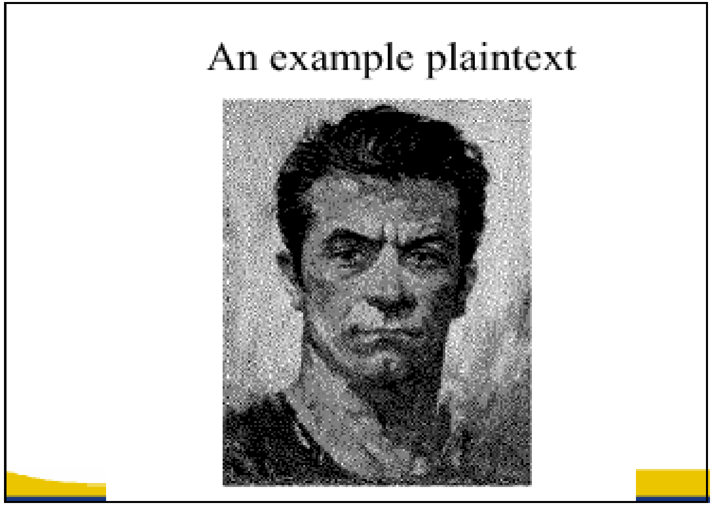
Cipher Block Chaining (CBC) mode encryption

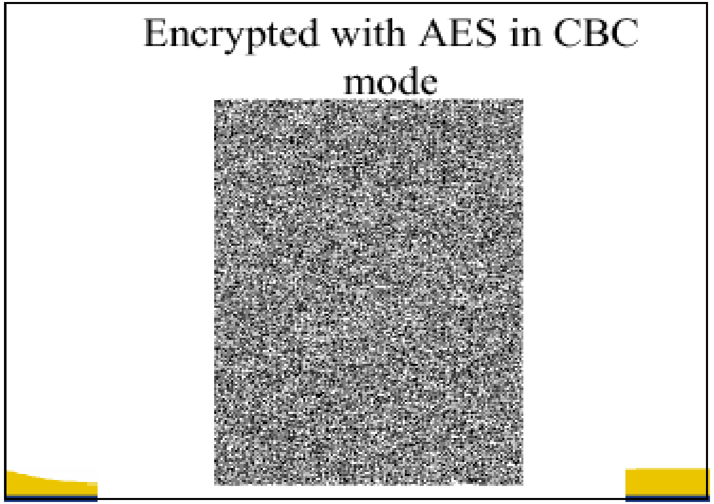
CBC decryption



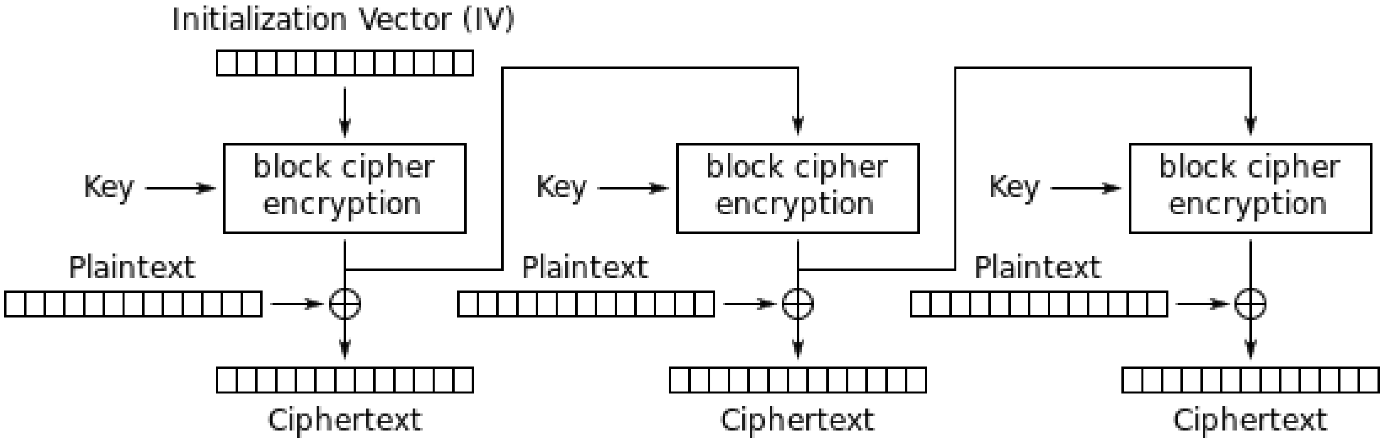
Cipher Block Chaining (CBC) mode decryption

Better





Output Feedback



Output Feedback (OFB) mode encryption

Security goals revisited

“Susceptibility to malicious insertions and modifications. Because each symbol is separately enciphered, an active interceptor who has broken the code can splice together pieces of previous messages and transmit a spurious new message that may look authentic.” - Phleeger & Phleeger in Security in Computing, Pearson, 2003

Is this a disadvantage of stream cipher? Why, why not?

Security goal of encryption: Confidentiality

Status

Confidentiality: Check!

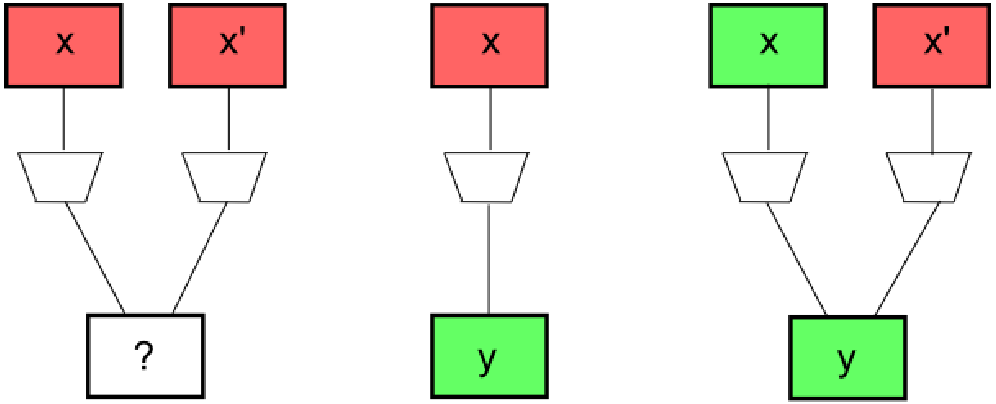
Integrity: Missing

Hands-on

```
[modes]$ cat myfile
1234567890abcde
1234567890abcde
1234567890abcde
1234567890abcde
[modes]$ openssl enc -aes-128-ecb -e -K 0 -in myfile -nopad | xxd
hex string is too short, padding with zero bytes to length
00000000: 4049 2e80 ddda bc83 0fa2 2096 1d47 c439  @I........ ..G.9
00000010: 4049 2e80 ddda bc83 0fa2 2096 1d47 c439  @I........ ..G.9
00000020: 4049 2e80 ddda bc83 0fa2 2096 1d47 c439  @I........ ..G.9
00000030: 4049 2e80 ddda bc83 0fa2 2096 1d47 c439  @I........ ..G.9
[modes]$ openssl enc -aes-128-cbc -e -K 0 -in myfile -nopad -iv 0 | xxd
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
00000000: 4049 2e80 ddda bc83 0fa2 2096 1d47 c439  @I........ ..G.9
00000010: b42e 0395 8128 e946 ea26 b84c 6c61 7f13  .....(.F.&.Lla..
00000020: 0b52 c084 f04e d7ba f39e 86e3 af54 cf64  .R...N.......T.d
00000030: 1bbc 9a11 8163 8b06 ba0a cdb9 1245 0b0a  .....c.......E..
[modes]$
```

Hash functions and Message authentication codes (MACs)

Cryptographic hash functions



Finding
Collision

Finding
Inversion

Finding
2nd Pre-image

Message authentication code

Goal: Provide integrity

Process

Choose a cryptographic hash funciton h : $\{0,1\}^x \rightarrow \{0,1\}^n$

Sender: Send h(m),m

Receiver: Calculate h(m) and verify it matches h(m)

Examples MD5 (n = 128), SHA-256 (n = 256)

Hash-based MAC (HMAC)

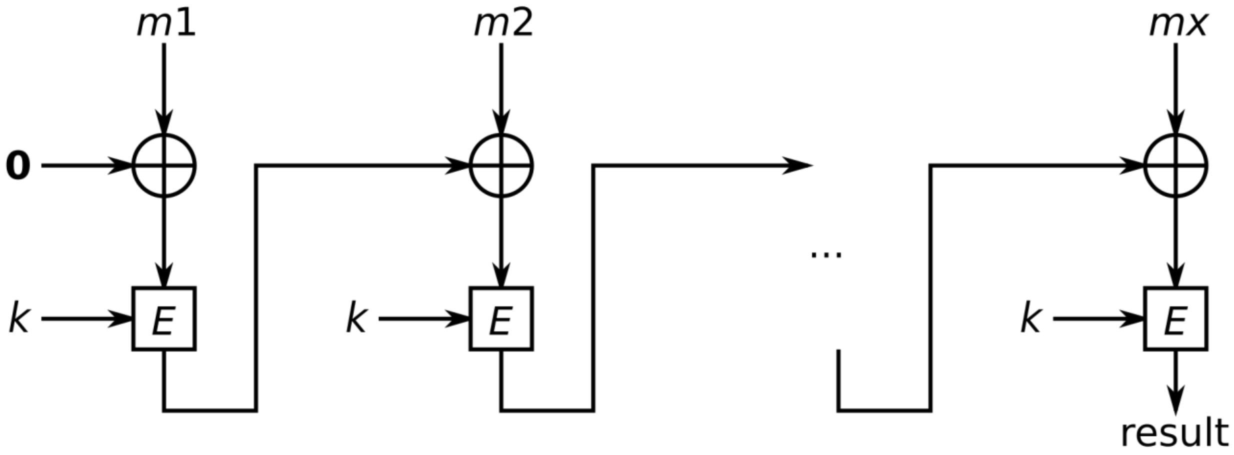
RFC2104: Hash-based MAC

HMAC(h,k,m) =

$$h\,(\,(k \oplus opad) \,\|\, h\,((k \oplus ipad) \,\|\, m)$$

HMAC provides integrity and authenticity

CBC-MAC



Car keys

Your car key sends the code for ”open the door”, together with a MAC, to the car whenever you press the button.

What could go wrong?

Replay attack: attacker records message and replays it later

We need some freshness: a timestamp or nonce

Hands-on

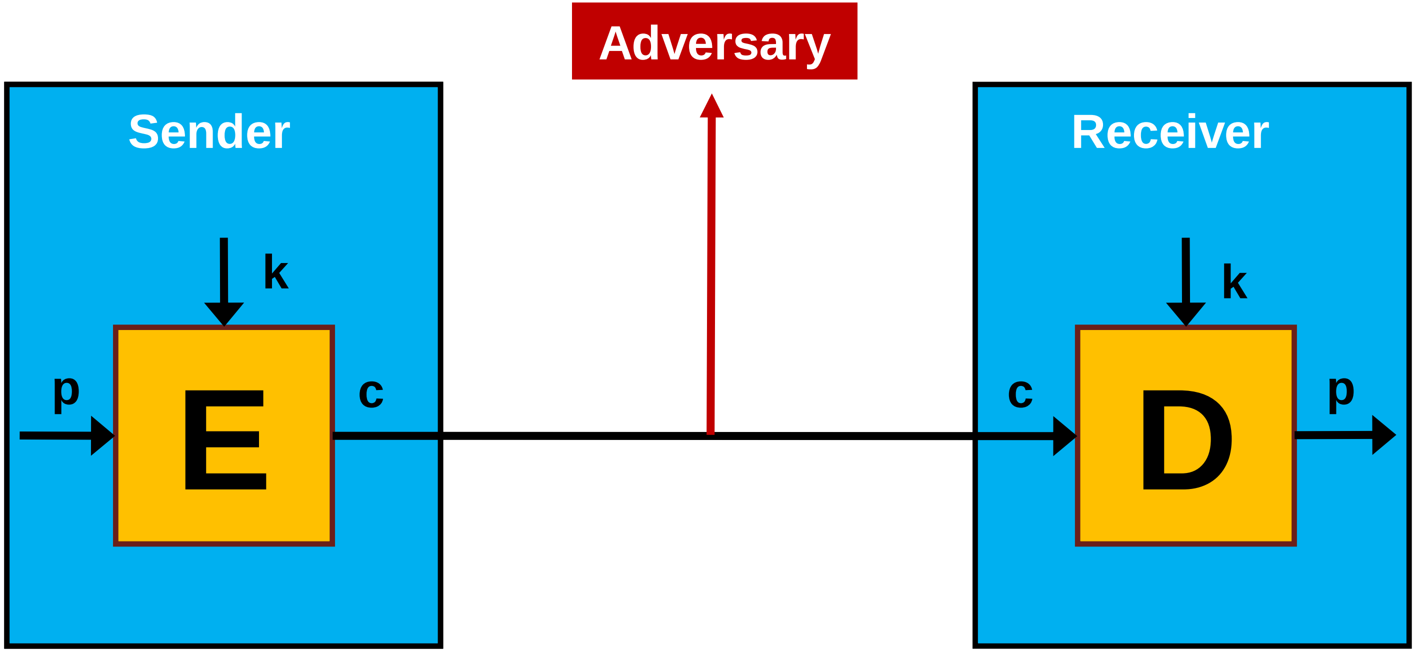
Terminal

File Edit View Search Terminal Help

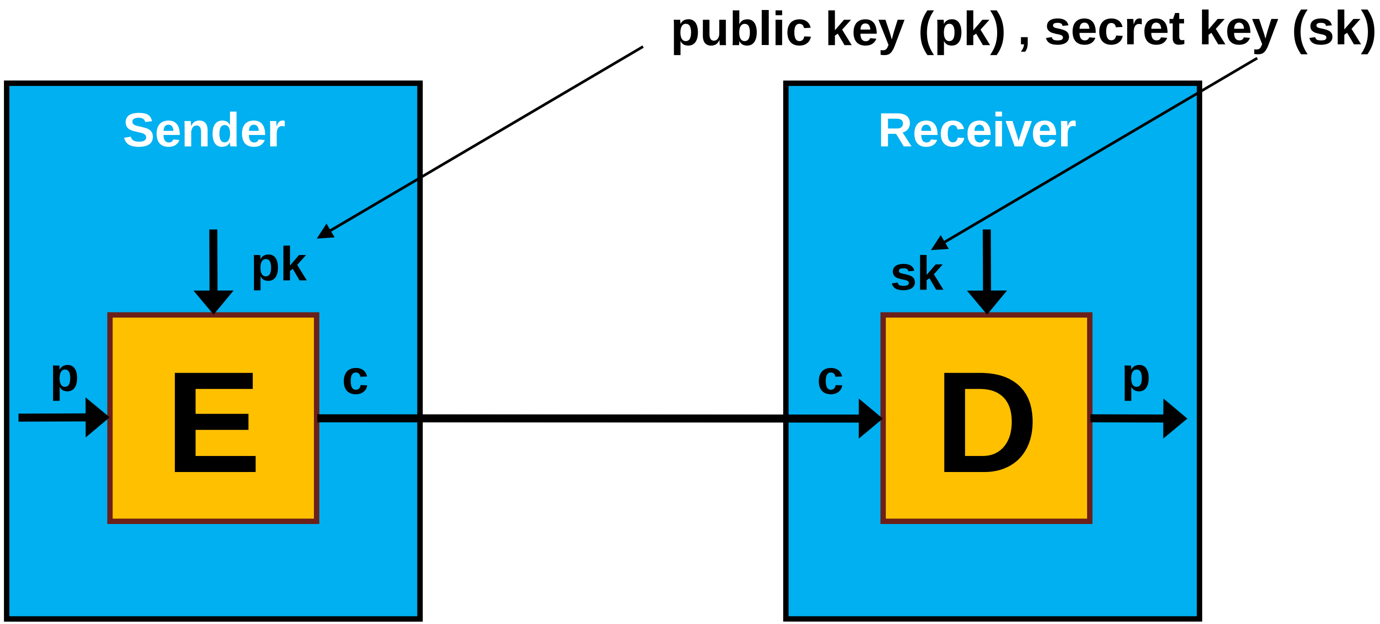
```
[hash]$ ls
message1.bin  message2.bin  pg11.txt
[hash]$ md5sum pg11.txt
eff1e5d84df1d3a543d1c578192a2367  pg11.txt
[hash]$ sha1sum pg11.txt
ddb2be9bdf09e20efc3b46dc670d50abec48f0e3  pg11.txt
[hash]$ md5sum message*
008ee33a9d58b51cfeb425b0959121c9  message1.bin
008ee33a9d58b51cfeb425b0959121c9  message2.bin
[hash]$ sha1sum message*
c6b384c4968b28812b676b49d40c09f8af4ed4cc  message1.bin
c728d8d93091e9c7b87b43d9e33829379231d7ca  message2.bin
[hash]$
```

Asymmetric ciphers

Cryptosystems



Enter: Asymmetric encryption



Analogy: Combination locks

Bob sends out locks with combination he only knows

Alice picks one of Bob's locks, places her message in a box and locks it with Bob's lock

Bob is the only one who can open the box now

No pre-shared key!

Bob

Publish public key, protect private key

Alice

Encrypt message with Bob’s public key

Bob

Decrypts with his private key

Rivest Shamir Adleman (RSA), 1978

First asymmetric cryptosystem

RSA encryption and decryption

Public key (N,e), private key (d)

$C = M^e \pmod{N}$

$M = C^d \pmod{N}$

Asymmetric encryption: Yes! But what about non-repudiaton?

Reverse it for Digital Signatures

Public key (N,e), private key (d)

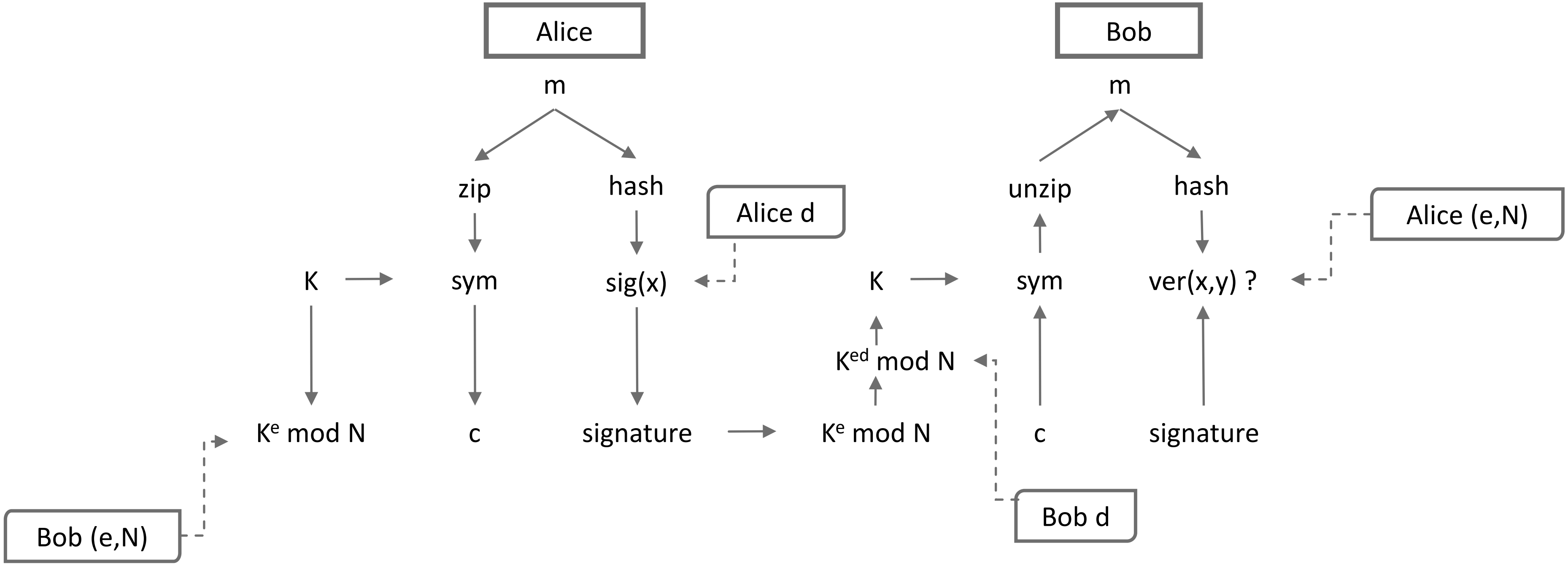
Signature sig(M) = M^d (mod N)

Verify ver(M,sig(M)) = true iff M = $(M^d)^e$ (mod N)

Hands-on

```
[rsa]$ bc
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Softwa
re Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
p=13
q=17
n=p*q
n
221

phi=(p-1)*(q-1)
phi
192

e=5
d=77
e*d % phi
1

m=123

c=m^e % n
c
106

c^d % n
123

m^(e*d)%n
123

quit
[rsa]$ openssl genrsa -out mykeys 2048
```

Putting it all togehter



Later, real-world crypto protocols

Wrap-up

Security goals achieved

Confidentiality

Integrity

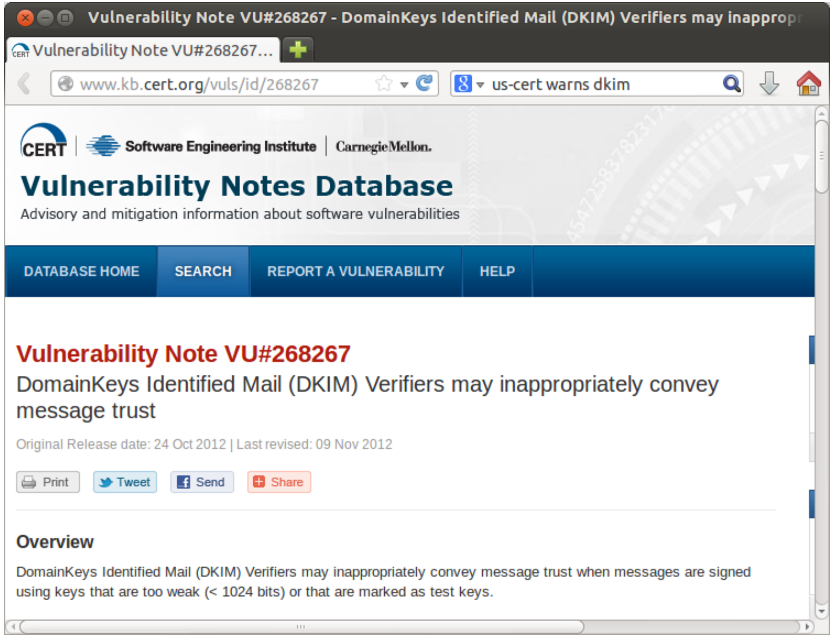
Authenticity

Non-repudiation

CHECK!

But crypto can still fail

Small keys fail



Collision fail

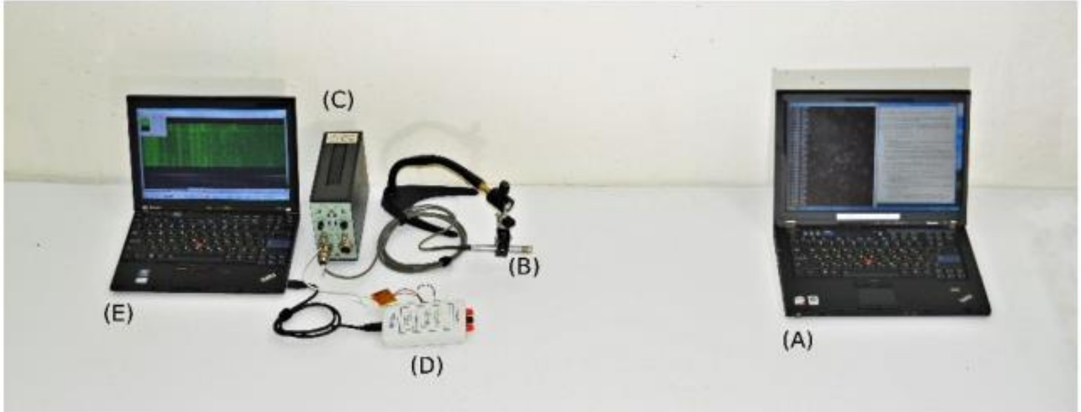
Impressive fail

New attack steals e-mail decryption keys by capturing computer sounds

Scientists use smartphone to extract secret key of nearby PC running PGP app.

by **Dan Goodin** - Dec 18, 2013 11:25 pm UTC

Share Tweet 108



Bad choice fail

IRS Encourages Poor Cryptography

Buried in one of the documents are the rules for encryption:

> While performing AES encryption, there are several settings and options depending on the tool used to perform encryption. IRS recommended settings should be used to maintain compatibility:
>
> - Cipher Mode: ECB (Electronic Code Book).
> - Salt: No salt value
> - Initialization Vector: No Initialization Vector (IV). If an IV is present, set to all zeros to avoid affecting the encryption.
> - Key Size: 256 bits / 32 bytes Key size should be verified and moving the key across operating systems can affect the key size.
> - Encoding: There can be no special encoding. The file will contain only the raw encrypted bytes.
> - Padding: PKCS#7 or PKCS#5.

ECB? Are they serious?

DIY fail

Smart grid security WORSE than we thought

OSGP's DIY MAC is a JOKE

11 May 2015 at 02:03, Richard Chirgwin 222 36 22

Backdoor fail

Topic: *Security*

Follow via:

NIST finally dumps NSA-tainted random number algorithm

Summary: *Many years since a backdoor was discovered, probably planted by the NSA, public pressure finally forces NIST to formally remove Dual_EC_DRBG from their recommendations.*

By Larry Seltzer for Zero Day | April 23, 2014 -- 14:04 GMT (07:04 PDT)

Follow @lseltzer

Comments 2 Vote 1 more +

Supply chain fail

Schneier on Security

Blog Newsletter Books Essays News Talks Academic About Me

Home > Blog

Crypto AG Was Owned by the CIA

The Swiss cryptography firm Crypto AG sold equipment to governments and militaries around the world for decades after World War II. They were owned by the CIA:

> But what none of its customers ever knew was that Crypto AG was secretly owned by the CIA in a highly classified partnership with West German intelligence. These spy agencies rigged the company's devices so they could easily break the codes that countries used to send encrypted messages.

This isn't really news. We have long known that Crypto AG was backdooring crypto equipment for the Americans. What is new is the formerly classified documents describing the details:

> The decades-long arrangement, among the most closely guarded secrets of the Cold War, is laid bare in a classified, comprehensive CIA history of the operation obtained by The Washington Post and ZDF, a German public broadcaster, in a joint reporting project.
>
> The account identifies the CIA officers who ran the program and the

Search

Powered by DuckDuckGo

Go

Blog Essays Whole site

Subscribe

About Bruce Schneier

Malware fail

EMNER *Cyberkrig, It-sikkerhed, Kryptering, Malware-virus* Se kommentarer (4)

Krypto-guru: Tiden er ved at løbe fra kryptering

Adi Shamir, en af opfinderne af RSA-kryptering, mener, at kryptografiens rolle til dels bliver udspillet af nye typer avanceret malware, som ofte stammer fra regeringer.

*Af **Mikkel Meister** Tirsdag, 5. marts 2013 - 16:16*

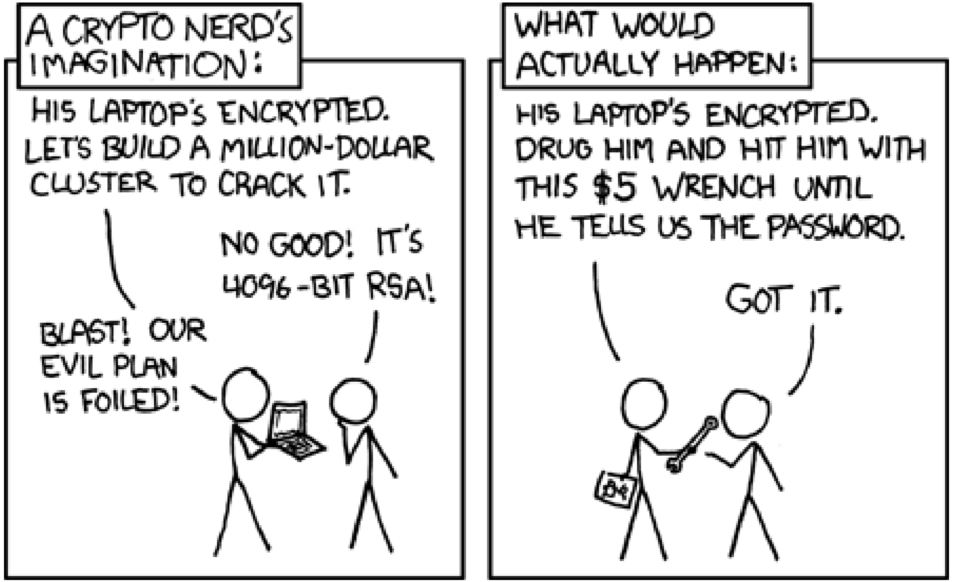
Kryptografi har ikke længere den samme vigtige rolle for it-sikkerhed som tidligere.

Det siger en af feltets helt store personligheder, Adi Shamir, som er en af skaberne af public key-kryptering, ifølge flere udenlandske it-medier.

Han henviser til højtprofilerede angreb de senere år, hvor selv de mest hærdede it-sikkerhedsstrukturer omgås af statssponsorerede hackere og avanceret malware som Stuxnet.

»Jeg mener afgjort, at kryptografi er ved at blive mindre afgørende. Selv de mest sikre it-systemer på de mest isolerede steder er blevet gennemhullet i de senere år af flere APT'er (Advanced Persistant Threat eller statsmalware, *red.*) og andre avancerede typer af angreb,« sagde Adi Shamir på RSA 2013-konferencen i sidste uge, skriver Threatpost.com.

Real-world fail



Suggested reading